Hybrid Single/Double Precision Floating-Point Computation on GPU Accelerators for 2-D FDTD

Hasitha Muthumala Waidyasooriya, Yasuhiro Takei, Masanori Hariyama and Michitaka Kameyama
Graduate School of Information Sciences, Tohoku University
Aoba 6-6-05, Aramaki, Aoba, Sendai, Miyagi, 980-8579, Japan
Email: {hasitha, takei, hariyama, kameyama}@ecei.tohoku.ac.jp

Abstract— *Acceleration of FDTD (Finite-Difference Time-Domain) is very important in computational electromagnetic. We propose a hybrid single/double precision floating-point computation to accelerate FDTD on GPUs. We apply single-precision when the dynamic range of the electromagnetic field is low and double-precision when the dynamic range is high. According to the experimental results, we achieved over 35 times of speed-up compared to the CPU implementation and over 1.79 times speed-up compared to the conventional GPU acceleration.*

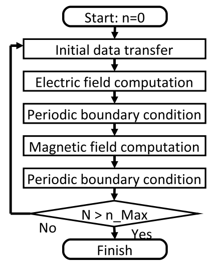
Keywords: GPGPU, FDTD, high-performance-computing

1. Introduction

Computational electromagnetic shows a rapid development recently due to the introduction of processors that have parallel processing capability such as multicore CPUs and GPUs (Graphic Processing Units). FDTD (Finite-Difference Time-Domain) algorithm [1] is one of the most popular method of computational electromagnetic simulation due to its simplicity and very high computational efficiency. It has been widely used in many applications such as coil modeling [2], resonance characteristics analysis [3], etc. Many of these applications require double precision floating-point computation to satisfy the stability condition [4].

There are many works that use GPUs [5] to accelerate FDTD. Such works consider how to parallelize the FDTD computation so as to use many nodes as possible. However, using more nodes means more cost and more power consumption. In this paper, we focus on extracting more performance from the same hardware by using high-precision computation only when it is necessary.

We consider the FDTD method used in resonance characteristics analysis of a cylindrical cavity [3] as the example application. We analyze the characteristics of the application and apply double-precision floating-point computation when the dynamic range is large and single-precision floating-point computation when the dynamic range is small. dynamic range refers to the ratio between the largest and the smallest values of electric (or magnetic) field. According to the experimental results, we achieved over 35 times of speed-up compared to the CPU implementation. This speed-up is almost 1.79 times of the conventional GPU acceleration.



(a) FDTD process

$$E_z^{n+1}(i,j) = E_z^n(i,j) - P_y(i,j)\left\{H_x^{n+\frac{1}{2}}(i,j+1/2) - H_x^{n+\frac{1}{2}}(i,j-1/2)\right\}$$
$$+ P_x(i,j)\left\{H_y^{n+\frac{1}{2}}(i+1/2,j) - H_y^{n+\frac{1}{2}}(i-1/2,j)\right\}$$
$$H_x^{n+\frac{1}{2}}(i,j+1/2) = H_x^{n-\frac{1}{2}}(i,j+1/2) - Q_y(i,j)\left\{E_z^{n+1}(i,j+1) - E_z^n(i,j)\right\}$$
$$H_y^{n+\frac{1}{2}}(i+1/2,j) = H_y^{n-\frac{1}{2}}(i+1/2,j) - Q_x(i,j)\left\{E_z^{n+1}(i+1,j) - E_z^n(i,j)\right\}$$

(b) FDTD computation

Fig. 1: FDTD

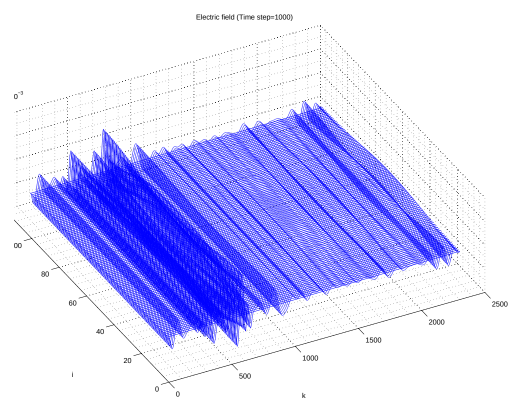


Fig. 2: Dynamic range of the electric field values

2. Hybrid floating-point computation

Figure 1(a) shows the FDTD algorithm [1]. It starts with an initial data of electric and magnetic fields. The initial data are processed to obtain the electric field for the first time step. After that, the boundary conditions are applied. Then the magnetic field data are obtained and the boundary conditions are applied. This process continues for n time steps. Figure 1(b) shows the electric and magnetic field computations. Electric and magnetic fields (in x, y, z directions) are denoted by E and H respectively. The coordinates of the 2D fields are denoted by (i, k).

To increase the speed-up, we observe the characteristics of our application [3]. According to [3], the electromagnetic field outside the cavity is weaker than that inside. Figure 2 shows the computation grid for FDTD calculation. The electric field far away from the grid origin has a small dynamic range. We observed similar characteristics from

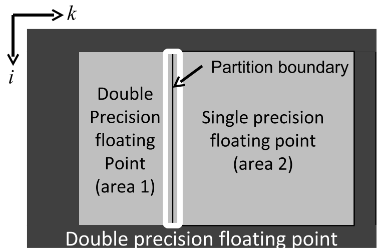


Fig. 3: Partition for single/double precision computation

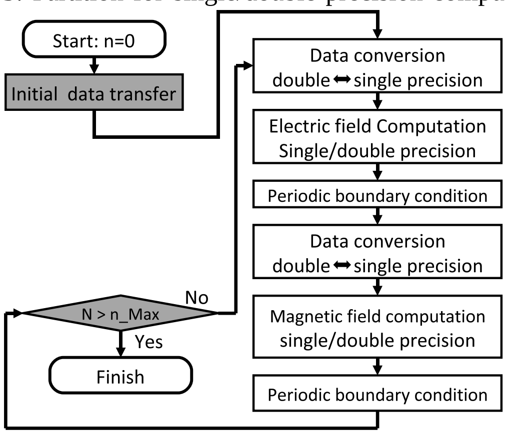


Fig. 4: Hybrid single/double precision computation

the magnetic field data analysis also. Therefore, we use double-precision floating-point when the dynamic range is large and single-precision floating-point when the dynamic range is small. As shown in Fig.3, the entire boundary and "area 1" are done in double precision floating-point while the rest of the computations in "area 2" are done in single precision floating-point. One problem in this method is the float-to-double conversion overhead. As shown in Fig.1(b), calculation of electric field of the coordinate (i, k) requires the magnetic fields at its left and right coordinates. Similarly, calculation of magnetic field of coordinate $(i, k + 1/2)$ requires the electric field at its left and right coordinates. To compute the electric and magnetic fields on the partition boundary in Fig.3, we need both double and single precision values that belong to area 1 and area 2 respectively. Therefore, we need float-to-double conversion and this is an additional overhead. Figure 4 shows the flow-chart of the hybrid single/double precision floating-point computation.

3. Evaluation

For the evaluation, we use Intel core i7 960 CPU and GeForce GTX 590 GPU. As shown in Fig.5, the proposed method is 1.79 times faster than the conventional double-precision GPU implementation [5]. Double-precision floating-point computation requires two CUDA cores while single-precision requires only one. This is the reason for the speed-up of the proposed method.

Figure 6 shows the processing time against the amount of double precision computation. Note that, the "double precision computation area" refers to the percentage of the grid area done in double-precision. The processing time increases with the amount of double precision computation. However, the processing time of "95% double precision computation" is larger than that of "100% double precision computation". This is due to the float to double conversion overhead in hybrid single/double precision computation.

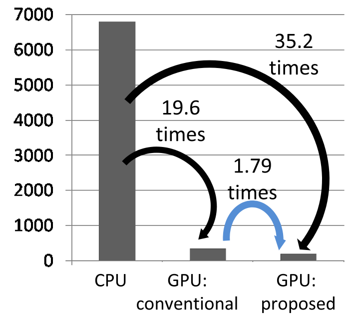


Fig. 5: Processing time comparison

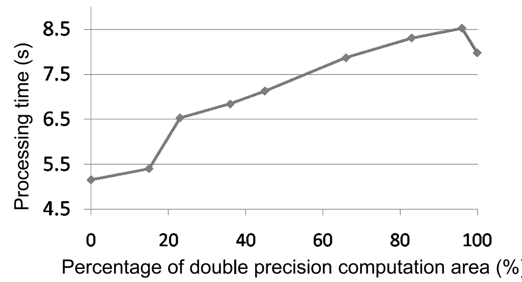


Fig. 6: Accuracy of the hybrid computation

4. Conclusion

In this paper, we proposed a hybrid single/double precision floating-point computation to accelerate FDTD on GPU. Since the amount of hardware required for double-precision is two times larger than that of the single-precision, we can increase the performance of the GPU by doing more computation in single-precision. However, due to the double-to-float conversion overhead, at least 85% of single-precision computation is needed for a considerable speed-up.

References

[1] H. S. Yee, "Numerical Solution of Initial Boundary Value Problems Involving Maxwell's Equations in Isotropic Media", IEEE Transactions on Antennas and Propagation, Vol.14, No.3, pp.302-307, 1966.

[2] T. S. Ibrahim,R. Lee, B. A. Baertlein, Y.Yu, and P.M. L.Robitaille, "Computational analysis of the high pass birdcage resonator: finite-difference time-domain simulations for high-field MRI", Magn. Reson. Imag., vol.18, no.7, pp.835-843, 2000.

[3] Y. Ohtera, S. Iijima and H. Yamada, "Cylindrical Resonator Utilizing a Curved Resonant Grating as a Cavity Wall", micromachines, pp.101-113, 2012.

[4] A. Taflove and M. E. Brodwin, "Numerical Solution of Steady-State Electromagnetic Scattering Problems Using the Time-Dependent Maxwell's Equations", IEEE Transactions on Microwave Theory and Techniques, Vol23. Issue 8, pp.623-630, 1975.

[5] Z. Bo, X. Zheng-hui, R. Wu, L. Wei-ming, S. Xin-qing, "Accelerating FDTD algorithm using GPU computing", International Conference on Microwave Technology & Computational Electromagnetics (ICMTCE), pp.410-413, 2011.